Instance space analysis for outlier detection

Sevvandi Kandanaarachchi, Mario A Muñoz, Kate Smith-Miles
Monash University & University of Melbourne
Overview


Extend Campos et al. [2016]

- Algorithm Selection
- Using Instance Space Analysis
Outlier detection

- It means different things to different people
- What is the definition of an outlier?
- Hawkins: an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism
- We focus on ground truth when labelling outliers
Ground truth as an outlier

- Blue dot - a security breach, red dots - normal activity
Basic Mechanism

- data in $\mathbb{R}^n$
- A mapping $f : \mathbb{R}^n \to \mathbb{R}$
- such that outliers have different $f(x)$ compared with other points
Due to different definitions and mechanisms
What is the best outlier detection method for my problem?
Algorithm Selection

Datasets
Unsupervised
Outlier
Detection 
Algorithms
Match-maker
Match-maker needs to know truth to learn how outlier definitions match with algorithm performance
To match we need to know the outlier labels

- Common in industry applications to have labeled data and develop methods to detect these outliers
Our experiment with real data

- 12000+ datasets
  - Around 200 sources, many variants
- 12 outlier detection methods
- Meta-features
  - Describe a dataset
  - Each dataset can be represented as a feature vector
  - A way to compare datasets
  - 178 meta-features
- Matching outlier methods with datasets
- A lot of time on the Monash Cluster
Datasets = Instances = \( I \in \{I, F, Y, A\} \)

- Campos et al. datasets (only \( \leq 5\% \) outliers)
- Goldstein-Uchida datasets from *A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data*
- Muñoz et al. classification datasets from *Instance Spaces for Machine Learning Classification*
  - From UCI repository
  - Prepare for outlier detection
  - Downsampling each class - several variants
  - Convert categorical attributes numerical
  - Remove duplicate observations
  - Attend to missing values
Meta-features $= F \in \{I, F, Y, A\}$

- **Simple features**
  - Number of obs., attributes, binary attributes, ...

- **Statistical features**
  - skewness, kurtosis, mean to sd ...

- **Information theoretic features**
  - entropy, mutual information, ...

- **Density based features**
  - DBSCAN, Kernel density estimates ...

- **Residual based features**

- **Graph based features**
Breakdown of features

<table>
<thead>
<tr>
<th>Feature category</th>
<th>Number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic: Simple, Statistical and Information theoretical</td>
<td>25</td>
</tr>
<tr>
<td>Density based</td>
<td>77</td>
</tr>
<tr>
<td>Residual based</td>
<td>35</td>
</tr>
<tr>
<td>Graph based</td>
<td>41</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>178</strong></td>
</tr>
</tbody>
</table>

We use outlier labels to compute some features.
Outlier Detection Methods = $A \in \{I, F, Y, A\}$

- **Distance based**
  - KNN
  - KNNW
  - ODIN

- **Density based**
  - LOF
  - LDF
  - LDOF
  - LOOP
  - COF
  - SIMLOF
  - KDEOS
  - INFLO

- **Angle based**
  - FAST ABOD
Evaluation metric = \( Y \in \{I, F, Y, A\} \)

- Area under ROC, Precision-Recall curve, Precision@n
- We use area under ROC as the evaluation metric \( Y \)
- To validate meta-features
  - Define good performance as Area under ROC > 0.8 \( \rightarrow \tilde{Y} \)
  - One model for each outlier detection method

![Diagram of meta-features and prediction process]
Validating the meta-features

<table>
<thead>
<tr>
<th>Outlier detection method</th>
<th>Default accuracy(%)</th>
<th>Prediction accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COF</td>
<td>75.58</td>
<td>83.48</td>
</tr>
<tr>
<td>FAST ABOD</td>
<td>67.77</td>
<td>86.07</td>
</tr>
<tr>
<td>INFLO</td>
<td>83.22</td>
<td>89.29</td>
</tr>
<tr>
<td>KDEOS</td>
<td>90.96</td>
<td>92.81</td>
</tr>
<tr>
<td>KNN</td>
<td>68.16</td>
<td>86.56</td>
</tr>
<tr>
<td>KNNW</td>
<td>67.13</td>
<td>86.13</td>
</tr>
<tr>
<td>LDF</td>
<td>75.65</td>
<td>85.28</td>
</tr>
<tr>
<td>LDOF</td>
<td>80.08</td>
<td>87.36</td>
</tr>
<tr>
<td>LOF</td>
<td>74.63</td>
<td>84.07</td>
</tr>
<tr>
<td>LOOP</td>
<td>77.19</td>
<td>85.88</td>
</tr>
<tr>
<td>ODIN</td>
<td>79.09</td>
<td>87.00</td>
</tr>
<tr>
<td>SIMLOF</td>
<td>75.85</td>
<td>85.21</td>
</tr>
</tbody>
</table>
Understanding strengths and weaknesses of algorithms

- Instance space methodology
- Visually represent the datasets and the algorithm performances
- Understand the relative strengths and weaknesses
Datasets $(I)$

- **Algorithms (A)**
  - **Output**
  - **Performance metric (Y)**

- **Meta-features (F)**
  - **Feature selection**
  - **Project to 2D**
  - **Instance space**
Feature selection

- Select 7 features from 178
- Discard features with a small number of unique values
- Discard features that are highly correlated with each other and un-correlated with performance
- Cluster the remaining features
- Select the best combination of features
Chosen features

- **Mean_Entropy_Attr (generic)**
  - Mean entropy of attributes
- **IQR_TO_SD_Ratio_95 (generic)**
  - 95% of IQR to Standard deviation ratio of attributes
- **OPO_Res_ResOut_Median (residual)**
  - Median proxi-outlier residuals / median non-PO residuals
- **OPO_Res_Out_Mean (residual)**
  - Mean of outlier residuals / mean of non-outlier residuals
- **OPO_Den_Out_95P (density)**
  - 95% of density of non-outliers / 95% of density of outliers
- **OPO_GDeg_PO_Mean (graph)**
  - Mean graph degree inner points / mean graph degree non-inner points
- **OPO_GDeg_Out_Mean (graph)**
  - Mean graph degree of outliers / mean graph degree of non-outliers
The projection

- PBLDR : Prediction Based Linear Dimensionality Reduction
- Finds a projection with most linear trends in algorithm performance and feature values

\[
Z = \begin{bmatrix}
-0.0862 & -0.2078 \\
0.1737 & 0.1845 \\
-0.0460 & -0.2847 \\
-0.0938 & -0.2025 \\
0.1202 & 0.0378 \\
0.1854 & -0.0822 \\
0.3543 & -0.1325
\end{bmatrix}^T
\begin{bmatrix}
\text{Mean_Entropy_Attr} \\
\text{IQR_TO_SD_95} \\
\text{OPO_Res_ResOut_Median} \\
\text{OPO_Res_Out_Mean} \\
\text{OPO_Den_Out_95P} \\
\text{OPO_GDeg_PO_Mean} \\
\text{OPO_GDeg_Out_Mean}
\end{bmatrix}
\]
An SVM

- Partition the instance space
- Train an SVM for each outlier method
  - Input: instance coordinates in 2D
  - Binary output: For each method does the instance elicit good performance from the method
- Break ties using prediction probability of the SVM
Instance space
How to use it

1. New dataset
2. Meta-features
3. Instance space model
4. Probabilities of algorithm success
Takeaway

• No outlier method is superior to all other methods

• Need to find the suitable method for a given problem

• We find the strengths and weaknesses via instance space methodology and predict regions of good performance
Thank you!

- R package *outselect* at https://github.com/sevvandi/outselect
- Datasets at https://monash.figshare.com/articles/Datasets12338zip/7705127/4